LICEO B. RUSSELL

Via IV Novembre 35 – 38023 Cles

PROGRAMMAZIONE DIDATTICA

Informatica

Classe **1D**

Liceo Scientifico – Opzione Scienze Applicate

DOCENTE
Prof. Mattia Marini

ANNO SCOLASTICO
2026/27

ORE ANNUE
66

ORE A CONSUNTIVO
/

COMPETENZE DA RAGGIUNGERE AL TERMINE DELL'ANNO SCOLASTICO	• Saper analizzare un problema e formulare una soluzione algoritmica, scomponendolo in istruzioni elementari e strutture di controllo. • Saper riconoscere e classificare i principali componenti hardware di un PC, individuando la funzione di ciascuno e il ruolo svolto nel sistema. • Comprendere il funzionamento a basso livello di un computer: bit, byte, memoria e processore. • Comprendere la logica di funzionamento di un computer e i principi della codifica binaria. • Saper utilizzare i più comuni strumenti di composizione tipografica e di marcatura dei testi (Markdown e LaTeX). • Saper utilizzare un foglio di calcolo e le sue funzionalità avanzate. • Avere familiarità con gli ambienti di sviluppo integrati (IDE), in particolare Visual Studio Code. • Conoscere i fondamenti della programmazione e sviluppare semplici programmi in Python. • Avere confidenza con le strutture dati di base: liste, stringhe e matrici. • Saper utilizzare le funzioni di base della libreria standard di Python.
METODOLOGIE DIDATTICHE	• Lezione frontale e dialogata, con coinvolgimento attivo della classe. • Esercitazioni in laboratorio, individuali o a coppie. • Lavori di gruppo basati su peer learning. • Problem solving guidato, con analisi collettiva degli errori. • Assegnazione e sviluppo di progetti da svolgere a casa. • Simulazioni delle prove di verifica. • Attività di recupero e consolidamento per gli studenti che incontrano difficoltà.
STRUMENTI E CRITERI DI VALUTAZIONE	• Verifiche scritte, pratiche e orali. • Sviluppo di progetti individuali o di gruppo. • Partecipazione attiva e qualità degli interventi. • Progressi rispetto ai livelli di partenza. • Capacità di analisi e di risoluzione dei problemi. • Correttezza, chiarezza e completezza del codice prodotto.
TESTI, MATERIALI E STRUMENTI ADOTTATI	• Ambiente di sviluppo integrato (IDE): Visual Studio Code. • LIM (Lavagna Interattiva Multimediale). • Presentazioni multimediali interattive, esercizi e appunti a cura del docente. • Sito web del docente. • Laboratorio di informatica con PC e connessione a Internet.

Modulo 1 – Introduzione all'informatica

Ore previste: 5

CONOSCENZE E/O CONTENUTI	• Concetto di algoritmo: definizione e applicazione a problemi quotidiani. • Funzionamento di un computer: bit, byte, memoria e processore. • Rappresentazione di numeri, caratteri e istruzioni in codifica binaria.

	• Cenni al sistema di numerazione binario e alle conversioni con il sistema decimale. • Panoramica dei principali componenti hardware di un computer. • Creazione di semplici algoritmi utilizzando il linguaggio naturale.
ABILITÀ E COMPETENZE	• Riconoscere e classificare i principali componenti hardware di un PC, individuando con chiarezza la funzione di ciascuno e il ruolo svolto nel sistema. • Comprendere il funzionamento a basso livello di un computer e costruire modelli mentali chiari del flusso dei dati e della rappresentazione astratta della memoria. • Rappresentare informazioni (numeri, caratteri e istruzioni) in forma binaria. • Sviluppare un'intuizione della logica algoritmica, intesa come capacità di costruire la soluzione di un problema a partire da un insieme ben definito di istruzioni elementari.

Modulo 2 – Introduzione alla logica della programmazione

Ore previste: 6

CONOSCENZE E/O CONTENUTI	• Distinzione tra linguaggi di programmazione ad alto e basso livello. • Concetto di compilatore e di interprete. • Sintassi di base del linguaggio Python. • Variabili e tipi di dato. • Operazioni aritmetiche e booleane di base. • Esecuzione condizionale: `if`, `elif`, `else` ed espressioni booleane. • Produzione dell'output a schermo tramite `print()`. • Errori di sintassi e di esecuzione: lettura e interpretazione dei messaggi. • Utilizzo di base dell'IDE Visual Studio Code.
ABILITÀ E COMPETENZE	• Scrivere semplici programmi in Python, utilizzando correttamente variabili, tipi di dato, operatori e strutture condizionali. • Utilizzare Visual Studio Code per scrivere, eseguire e correggere programmi. • Individuare e correggere semplici errori di sintassi e di esecuzione.

Modulo 3 – Introduzione a Python e VS Code

Ore previste: 25

CONOSCENZE E/O CONTENUTI	• Cicli `while`, concetto di iterazione e utilizzo di `break`. • Liste: concetto di struttura dati e funzionalità di base. • Stringhe: funzionalità di base. • Semplici algoritmi su liste e stringhe: ricerca, conteggio e accumulo.
ABILITÀ E COMPETENZE	• Acquisire familiarità con i concetti di iterazione e con le strutture dati di base, sapendo utilizzare tali elementi in semplici programmi Python. • Riconoscere i modelli ricorrenti (pattern) della programmazione e saperli riadattare a problemi analoghi. • Risolvere semplici problemi utilizzando liste e stringhe.

Modulo 4 – Concetti avanzati di programmazione in Python

Ore previste: 15

CONOSCENZE E/O CONTENUTI	• Cicli `for` e utilizzo di `range()`. • Iterazione su strutture dati iterabili. • Funzioni della libreria standard di Python: `len()`, `max()`, `min()`, `sum()`, `sorted()`, `reversed()`, `enumerate()`. • Acquisizione dell'input da tastiera e conversione dei tipi di dato; scrittura di semplici programmi interattivi. • Matrici: rappresentazione tramite liste di liste, operazioni di base e algoritmi di manipolazione.
ABILITÀ E COMPETENZE	• Scrivere programmi Python con maggiore rapidità e sicurezza, utilizzando le funzionalità più avanzate del linguaggio. • Sviluppare programmi più complessi sfruttando costrutti idiomatici. • Operare con strutture dati avanzate, in particolare matrici. • Gestire l'input dell'utente e convertire correttamente i tipi di dato, sviluppando semplici applicazioni interattive.

Modulo 5 – Foglio di calcolo e basi di composizione tipografica

Ore previste: 15

CONOSCENZE E/O CONTENUTI	**IL FOGLIO DI CALCOLO** • Aritmetica tra celle e modalità di riferimento relativo e assoluto. • Funzioni statistiche fondamentali (ad es. `SOMMA`, `MEDIA`, `CONTA.VALORI`). • Funzioni logiche (ad es. `SE`, `E`, `O`). • Filtri di base e ordinamento di dati tabellari. • Cenni di visualizzazione dei dati e generazione di grafici a partire da dati tabellari. **COMPOSIZIONE TIPOGRAFICA** • Concetto di composizione tipografica (typesetting) e scrittura di documenti in modalità WYSIWYM. • Struttura di un documento in Markdown e/o LaTeX. • Elementi principali del documento: sezioni, elenchi puntati e numerati, tabelle, immagini, link e formule matematiche. **CENNI DI ANALISI DI DATI** • Lettura di file di testo e di dati in formato `CSV` (cenni). • Generazione automatica di report tramite Python e Markdown.
ABILITÀ E COMPETENZE	• Saper utilizzare le funzioni di base dei fogli di calcolo, sapendo organizzare calcoli automatici e generare semplici grafici a partire da dati tabellari. • Saper redigere semplici documenti in Markdown e LaTeX, con particolare attenzione ai documenti scientifici e alle tabelle. • Maturare una visione orientata all'automazione dei processi di analisi dei dati, riconoscendo i vantaggi rispetto alle procedure manuali.